What is claimed is:

1. A method for performing remote access commands between nodes, the method comprising:

issuing a first request from a first node to a second node, the first request requesting a data access task be performed between the first node and the second node;

receiving, at the first node, a first response from the second node that partially completes the data access task;

issuing at least one subsidiary request from the first node to the second node to further complete the data access task between the first node and the second node, the at least one subsidiary request based on an amount of partial completion of the data access task between the first node and the second node; and

receiving, from the second node in response to the at least one subsidiary request, at least one corresponding subsidiary response that further completes the data access task between the first node and the second node.

2. The method of claim 1 wherein issuing the first request from the first node comprises:

detecting an application request in a request queue, the application request identifying the data access task to be performed between the first and second node; and

wherein the method comprises:

repeating issuing at least one subsidiary request and receiving at least one corresponding subsidiary response between the first and second nodes until the data access task is totally complete between the first and second nodes.

3. The method of claim 2 comprising:

allocating context resources in the first node for receipt of the first response and for receipt of the at least one subsidiary response, the context resources allocated to support receipt of the first response and for receipt of the at least one subsidiary response that contain data in an amount not exceeding a data allotment to support at least partial completion of the data access task.

4. The method of claim 3 comprising:

pre-empting the context resources in the first node allocated for receipt of the first response and for receipt of at least one subsidiary response prior to full completion of the data access task;

issuing a second request from the first node by re-activating the context resources that were pre-empted.; and

at a time after issuance of the second request from the first node, allocating context resources to support at least partial completion of the data access task in order to continue repeating issuing at least one subsidiary request and receiving at least one corresponding subsidiary response between the first and second nodes until the data access task is totally complete between the first and second nodes

5. The method of claim 4 wherein allocating context resources to support at least partial completion of the data access task comprises:

reserving the context resources to process subsidiary responses containing data limited in amount according to a preset data allotment identifying an amount of data to be transferred between the first and second nodes.

6. The method of claim 1 wherein the second request is at least one of:

a different type of data access request than the first request; and

received in a request queue that is different that a request queue from which the first request was received.

7. The method of claim 4 wherein issuing at least one subsidiary request comprises:

calculating a remaining amount of data required to complete the data access task between the first node and the second node; and

creating at least one subsidiary request to reference at least a portion of the remaining amount of data required to complete the data access task.

8. The method of claim 7 wherein calculating the remaining amount of data comprises:

determining a total completed amount of data processed for the data access task by the first request and associated first response and all subsidiary requests and corresponding subsidiary responses between the first and second node; and

determining the remaining amount of data required to complete the data access task as a difference between an initial amount of data specified by an application request and the total completed amount of data.

9. The method of claim 4 wherein:

the first and second nodes are Infiniband nodes that utilize Infiniband channel adapters to exchange the first request and the at least one subsidiary request and the corresponding first response and the at least one subsidiary response;

the application request is a remote direct memory access request for the first node to access data in a memory at the second node; and

the initial amount of data specified by the application request is a total amount of data that the first node is to access in the memory at the second node.

10. The method of claim 9 wherein the first request and the at least one subsidiary request are Infiniband read remote direct memory access commands issued by the first node to read data in the memory from the second node.

11. The method of claim 1 comprising:

establishing a data allotment as a maximum amount of data to be used when responding to requests to transfer portions of data between the nodes, such that if a total amount of data to be transferred between the first node and the second node is greater than the data allotment, the second node provides the first response and the at least one subsidiary response that contain response data that does not exceed the data allotment; and

allocating context resources in the first node for receipt of the first response and for receipt of the at least one subsidiary response, the context resources allocated to support receipt of responses in an amount that does not exceed the data allotment.

12. The method of claim 11 wherein establishing the data allotment comprises:

dynamically determining the data allotment between the first and second nodes based on at least one external data allotment event, such that if the at least one external data allotment event occurs, the first and second nodes change a value of the data allotment.

13. A communications interface comprising:

a processor;

a communications port; and

an interconnection mechanism coupling the processor and the communications port;

wherein the processor executes logic of a communications interface application to form a communications interface process that performs remote access commands between nodes by performing the operations of:

issuing a first request from a first node to a second node over the communications port, the first request requesting a data access task be performed between the first node and the second node;

receiving, over the communications port at the first node, a first response from the second node that partially completes the data access task;

issuing, over the communications port, at least one subsidiary request from the first node to the second node to further complete the data access task between the first node and the second node, the at least one subsidiary request based on an amount of partial completion of the data access task between the first node and the second node; and

receiving, over the communications port from the second node in response to the at least one subsidiary request, at least one corresponding subsidiary response that further completes the data access task between the first node and the second node.

14. The communications interface of claim 13 wherein when the communications interface application performs the operation of issuing the first request from the first node, the communications interface application performs the operations of:

detecting an application request in a request queue in the communications interface, the application request identifying the data access task to be performed between the first and second node; and

wherein the method comprises:

repeating issuing at least one subsidiary request and receiving at least one corresponding subsidiary response between the first and second nodes until the data access task is totally complete between the first and second nodes.

15. The communications interface of claim 14 wherein the communications interface application performs the operation of:

allocating context resources in the first node for receipt of the first response and for receipt of at least one subsidiary response, the context resources allocated to support receipt of the first response and for receipt of the at least one subsidiary response that contain data in an amount not exceeding a data allotment to support at least partial completion of the data access task.

16. The communications interface of claim 15 wherein the communications interface application performs the operations of:

pre-empting the context resources in the first node allocated for receipt of the first response and for receipt of at least one subsidiary response prior to full completion of the data access task;

issuing a second request from the first node by re-activating context resources that were pre-empted; and

at a time after issuance of the second request from the first node, allocating context resources to support at least partial completion of the data access task in order to continue repeating issuing at least one subsidiary request and receiving at least one corresponding subsidiary response between the first and second nodes until the data access task is totally complete between the first and second nodes

17. The communications interface of claim 16 wherein when the communications interface application performs the operation of allocating context resources in an amount

capable of supporting at least partial completion of the data access task, the communications interface application performs the operation of:

reserving the context resources to process subsidiary responses containing data limited in amount according to a preset data allotment identifying an amount of data to be transferred between the first and second nodes.

18. The communications interface of claim 13 wherein the second request is at least one of:

a different type of data access request than the first request; and

received in a request queue that is different that a request queue from which the first request was received.

19. The communications interface of claim 17 wherein when the communications interface application performs the operation of issuing at least one subsidiary request, the communications interface application performs the operations of:

calculating a remaining amount of data required to complete the data access task between the first node and the second node; and

creating the at least one subsidiary request to reference at least a portion of the remaining amount of data required to complete the data access task.

20. The communications interface of claim 19 wherein when the communications interface application performs the operation of calculating the remaining amount of data, the communications interface application performs the operations of:

determining a total completed amount of data processed for the data access task by the first request and associated first response and all subsidiary requests and corresponding subsidiary responses between the first and second node; and

determining the remaining amount of data required to complete the data access task as a difference between an initial amount of data specified by an application request and the total completed amount of data.

21. The communications interface of claim 16 wherein:

the first and second nodes are Infiniband nodes that utilize Infiniband channel adapters to exchange the first request and the at least one subsidiary request and the corresponding first response and the at least one subsidiary response;

the application request is a remote direct memory access request for the first node to access data in a memory at the second node; and

the initial amount of data specified by the application request is a total amount of data that the first node is to access in the memory at the second node.

22. The communications interface of claim 21 wherein the first request and the at least one subsidiary request are Infiniband read remote direct memory access commands issued by the first node to read data in the memory from the second node.

23. The communications interface of claim 13 wherein the communications interface application performs the operations of:

establishing a data allotment in the communications interface, the data allotment being a maximum amount of data to be used when responding to requests to transfer portions of data between the nodes, such that if a total amount of data to be transferred between the first node and the second node is greater than the data allotment, the second node provides the first response and the at least one subsidiary response that contain response data that does not exceed the data allotment; and

allocating context resources in the first node for receipt of the first response and for receipt of the at least one subsidiary response, the context resources allocated to support receipt of responses in an amount that does not exceed the data allotment.

24. The communications interface of claim 23 wherein when the communications interface application performs the operation of establishing the data allotment the communications interface application performs the operation of:

dynamically determining the data allotment between the first and second nodes based on at least one external data allotment event, such that if the at least one external data allotment event occurs, the first and second nodes change a value of the data allotment.

25. A method for performing Infiniband read remote access commands between Infiniband nodes, the method comprising:

issuing a first read remote access command from a first Infiniband node to a second Infiniband node;

receiving at least one first response from the second Infiniband node that partially completes a data access task associated with the first remote access command;

pre-empting context resources associated with issuance of the first remote access command from the first Infiniband node prior to completion of the task associated with the first remote access command;

issuing a second remote access command from the first Infiniband node using the pre-empted context resources; and

issuing, from the first Infiniband node to the second Infiniband node, a series of subsidiary remote access commands derived from the first remote access command to receive corresponding subsidiary responses from the second Infiniband node to complete the data access task associated with the first read remote access command.

26. A communications interface comprising:

a processor;

a communications port; and

an interconnection mechanism coupling the processor and the communications port;

wherein the processor executes logic of a communications interface application to form a communications interface process that, when executed, provides a means, within the communications interface to performs remote access commands between nodes, the means including:

means for issuing a first request from a first node to a second node over the communications port, the first request requesting a data access task be performed between the first node and the second node;

means for receiving, over the communications port at the first node, a first response from the second node that partially completes the data access task;

means for issuing, over the communications port, at least one subsidiary request from the first node to the second node to further complete the data access task between the first node and the second node, the at least one subsidiary request based on an amount of partial completion of the data access task between the first node and the second node; and

means for receiving, over the communications port from the second node in response to the at least one subsidiary request, at least one corresponding subsidiary response that further completes the data access task between the first node and the second node.

27. A computer program product having a computer-readable medium including communications interface application computer program logic encoded thereon that, when performed in a communications interface having a coupling of a communications port and a processor provides a communications interface process that performs remote access commands between nodes via the operations of:

issuing a first request from a first node to a second node over the communications port, the first request requesting a data access task be performed between the first node and the second node;

receiving, over the communications port at the first node, a first response from the second node that partially completes the data access task;

issuing, over the communications port, at least one subsidiary request from the first node to the second node to further complete the data access task between the first node and the second node, the at least one subsidiary request based on an amount of partial completion of the data access task between the first node and the second node; and

receiving, over the communications port from the second node in response to the at least one subsidiary request, at least one corresponding subsidiary response that further completes the data access task between the first node and the second node.